

Parallel netCDF: A High Performance API for NetCDF File Access

Overview

Parallel netCDF (PnetCDF) is a library providing high-performance I/O while still maintaining file-format compatibility with Unidata's NetCDF.

NetCDF gives scientific programmers a space-efficient and portable means for storing data. However, it does so in a serial manner, making it difficult to achieve high I/O performance. By making some small changes to the API specified by NetCDF, we can use MPI-IO and its collective operations.

- [Download](#) has the latest release and development links as well as information about svn access.
- [Documentation](#): a [QuickTutorial](#), plus papers, presentations, articles, and other resources
- [Benchmarking](#): tools and suggestions for evaluating pnetcdf performance

News

- In the 1.3.0 release, the unsigned and 64-bit integer data types are supported for CDF-5 format. The unsigned data types include NC_UBYTE, NC_USHORT, NC_UINT, and NC_UINT64. The 64-bit integer data types are NC_INT64 and NC_UINT64.
- APIs for the new data types are supported. For C, they are `ncmpi_(i)put/(i)get_var*_ushort/uint/longlong/ulonglong`. For Fortran, they are `nfmpi_(i)put/(i)get_var*_int8`.
- A new set of "buffered"-put APIs is supported in 1.3.0 release. The nonblocking `iput/iget` APIs require the contents of user buffers not to be changed until the wait call completed. The `bput` APIs use a user attached buffer to make a copy of request data, so the user buffer is free to change once the `bput` call returns.
- The special character set, "special2", and multi-byte UTF-8 encoded characters introduced in the CDF-2 file format for variable, dimension, and attribute name strings are now supported.
- A set of example programs and [QuickTutorial](#) are now available.
- Nonblocking I/O is redesigned in the 1.2.0 release. It defers the I/O requests until "wait" call, so small requests can be aggregated into large ones for better performance.
- Two new hints, `nc_header_align_size` and `nc_var_align_size`, are added. The former allows pre-allocation of a larger header size to accommodate new header data in case new variables or attributes are added later. The latter aligns the starting file offsets of non-record variables. Refer to [VariableAlignment](#) for a more detailed description.
- Data consistency control has been revised. A more strict consistency can be enforced by using NC_SHARE mode at the file open/create time. In this mode, the file header is synchronized to the file if its contents have changed. Such file synchronization of calling `MPI_File_sync()` happens in many places, including `ncmpi_enddef()`, `ncmpi_redef()`, all APIs that change global or variable attributes, dimensions, and number of records.
- As calling `MPI_File_sync()` is very expensive on many file systems, users can choose more relaxed data consistency, i.e. by not using NC_SHARE. In this case, file header is synchronized among all processes in memories. No `MPI_File_sync()` will be called if header contents have changed. `MPI_File_sync()` will only be called when switching data mode, i.e. `ncmpi_begin_indep_data()` and `ncmpi_end_indep_data()`.

A note about Large File Support

As of Pnetcdf-0.9.2, we ship with support for CDF-2 formatted data. With this format, even 32 bit platforms can create netcdf datasets (files) greater than 2GB in size. See the file README.large_files in the source tree for more information. CDF-2 also allows more special characters in the name strings of defined dimension, variables, and attributes.

The maintainers of the serial NetCDF library added support for the CDF-2 format in netcdf-3.6.0. The support was based largely on work from Greg Sjaardema.

The CD-5 file format specification: supports unsigned and 64-bit integer data types and variables with more than 2^{32} array elements.

The CDF (or CDF-1) file format specification has been in use through netCDF library version 3.5.1.

File and Variable Limits

Both PnetCDF and NetCDF share limitations on file and variable sizes. More information can be found on the [FileLimits](#) page.

Required Software

PnetCDF requires an MPI implementation with MPI-IO support. Most MPI libraries have this nowadays. A parallel file system would also go a long way towards achieving highest performance.

Related Projects

PnetCDF makes use of several other technologies.

- ROMIO, an implementation of MPI-IO, provides optimized collective and noncontiguous operations. It also provides an abstract interface for a large number of parallel file systems.
- One of those file systems ROMIO supports is PVFS, a high performance parallel filesystem for linux clusters.

Today, there are several options for high level I/O libraries. Here are some discussions on the role of PnetCDF in this ecosystem:

- [pnetcdf_vs_hdf5?](#)
- [pnetcdf_vs_netcdf4?](#)

Mailing List

We discuss the design and use of the PnetCDF library on the `parallel-netcdf@mcs.anl.gov` mailing list. Anyone interested in developing or using parallel-netcdf is encouraged to join. Visit [the list information page](#) for details.

The URL for the list archive is <http://lists.mcs.anl.gov/pipermail/parallel-netcdf/>. You can browse even older mailing list messages at the older [mailing list archives](#)

Project Members

- Rob Latham, Rob Ross, and Rajeev Thakur (Argonne National Lab)
- Wei-keng Liao, Seung Woo Son, and Alok Choudhary (Northwestern University)
- Kui Gao (formally postdoc at Northwestern, now Dassault Systèmes Simulia Corp.)
- Jianwei Li (Northwestern, graduated in 2006)
- Bill Gropp (formerly ANL, now UIUC)

Citations

When referring to the Parallel netCDF project, please use our "permanent" URL:
`www.mcs.anl.gov/parallel-netcdf`. The 'trac' or 'www-unix' URLs could change.

If you are looking for a reference to use in a published paper, please cite our SC2003 paper

Acknowledgements

Parallel netCDF is sponsored by the Scientific Data Management Center (SDM) under the DOE program of Scientific Discovery through Advanced Computing (SciDAC).